

Urban Region Pre-training and Prompting: A Graph-based Approach

Jiahui Jin
jjin@seu.edu.cn
School of Computer Science and
Engineering, Southeast University
Nanjing, China

Yifan Song
yifansong2000@seu.edu.cn
School of Computer Science and
Engineering, Southeast University
Nanjing, China

Dong Kan
dongkan@seu.edu.cn
School of Computer Science and
Engineering, Southeast University
Nanjing, China

Haojia Zhu
zhuhaojia@seu.edu.cn
School of Computer Science and
Engineering, Southeast University
Nanjing, China

Xiangguo Sun
xiangguosun@cuhk.edu.hk
Department of Systems Engineering
and Engineering Management,
The Chinese University of Hong Kong
Hong Kong, China

Zhicheng Li
lizhicheng@seu.edu.cn
School of Computer Science and
Engineering, Southeast University
Nanjing, China

Xigang Sun
xigangsun@seu.edu.cn
School of Computer Science and
Engineering, Southeast University
Nanjing, China

Jinghui Zhang
jhzhzhang@seu.edu.cn
School of Computer Science and
Engineering, Southeast University
Nanjing, China

Abstract

Urban region representation is crucial for various urban downstream tasks. However, despite the proliferation of methods and their success, acquiring general urban region knowledge and adapting to different tasks remains challenging. Existing work pays limited attention to the fine-grained functional layout semantics in urban regions, limiting their ability to capture transferable knowledge across regions. Further, inadequate handling of the unique features and relationships required for different downstream tasks may also hinder effective task adaptation. In this paper, we propose a Graph-based Urban Region Pre-training and Prompting framework (GURPP) for region representation learning. Specifically, we first construct an urban region graph and develop a subgraph-centric urban region pre-training model to capture the heterogeneous and transferable patterns of entity interactions. This model pre-trains knowledge-rich region embeddings using contrastive learning and multi-view learning methods. To further refine these representations, we design two graph-based prompting methods: a manually-defined prompt to incorporate explicit task knowledge and a task-learnable prompt to discover hidden knowledge, which enhances the adaptability of these embeddings to different tasks. Extensive experiments on various urban region prediction tasks

and different cities demonstrate the superior performance of our framework.

CCS Concepts

• **Information systems** → *Data mining*; • **Computing methodologies** → Knowledge representation and reasoning.

Keywords

Urban Foundation Model, Region Representation, Graph Prompt

1 Introduction

An urban region is a geographical area comprising diverse spatial entities (e.g., shopping centers and roads) and their interactions (e.g., nearby and located in). Recently, urban region representation learning, which aims to extract vector embeddings from these entities, has become a significant research focus [65]. Effective urban region representations can be applied to various tasks, such as region popularity prediction [13, 25, 50, 69], house price prediction [48], and crime prediction [48, 50, 64, 69], enabling urban planners and businesses to enhance decision-making processes.

However, the complexity of urban environments and the diversity of applications pose significant challenges. **On the one hand, urban regions often appear similar in their spatial structures and functional layouts**, because the interactions among entities within these regions follow similar patterns. These shared patterns serve as generic knowledge across urban areas, facilitating the development of generalizable region representation models. As illustrated in Figure 1, Regions A and B showcase typical urban regions. In these examples, spatial structures are demonstrated by the proximity of residences to parks and shops, which enhances community convenience and accessibility. Functional layouts are characterized by the clustering of similar entities within areas, such

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD '25, August 3–7, 2025, Toronto, ON, Canada

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN XXX-X-XXXX-XXXX-X/XXXX/XX

<https://doi.org/XX.XXXX/XXXXXXXXXXXXXX>

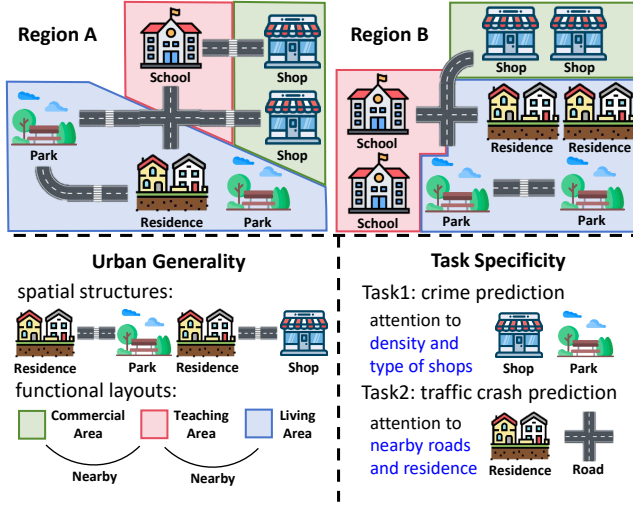


Figure 1: Example of urban generality and task specificity.

as shops in commercial areas and schools in teaching areas, and the close proximity of these distinct areas to each other. Therefore, an effective urban region representation model needs to capture the heterogeneous and generic patterns of interactions among entities within regions. **On the other hand, despite the broad similarities, each urban region and downstream task possesses unique details.** Specific tasks may require attention to different features within urban data. For example, in traffic crash prediction, the density of roads and the number of residents living nearby might be critical features, as they directly influence the probability of accidents occurring. Meanwhile, crime prediction models might focus on the density and type of POIs (Point-of-Interests) within the region, such as bars or late-night convenience stores. As a consequence, the balance between generality and specificity raises a crucial question: *Can we design an urban region representation framework that effectively integrates generalizable knowledge while accommodating the specific needs of distinct tasks?*

To solve this problem, this paper presents a simple yet effective framework with pre-training and prompting on geographical data, since this new paradigm has shown remarkable success in balancing embedding generality and task specificity in the areas like natural language processing [29, 39], computer vision [67, 68], and graphs [43, 44]. The intrinsic reason behind this superiority, as many researchers reported [49], lies in the powerful capability of data manipulation from the prompt side, and knowledge learning from the pre-training side, which can offer flexible adaptation for various downstream tasks and useful knowledge for generalizable representation. However, handling such complex non-linear geographical data requires prior experience. Challenges are twofold:

- **Difficulties in capturing the heterogeneous and generic knowledge from complex spatial entity interactions.** Existing methods mostly predefine aggregations and statistics of some urban entities and relationships, and then use these predefined features for downstream tasks [24, 55, 63], which often fail to capture the spatial structures and functional layouts between various entities within regions.

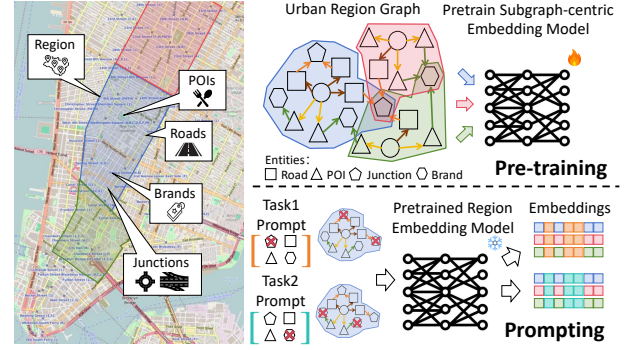


Figure 2: Main idea of urban pre-training and prompting.

- **Difficulties in capturing task-specific patterns.** Existing methods usually use simple prompts like directly concatenating them to the pre-trained embeddings [54, 69], which might work for certain domains but might fall short for urban region tasks with heterogeneous and complex geographical entity interactions.

In response to these challenges, we propose GURPP, a Graph-based Urban Region Pre-training and Prompting Framework as shown in Figure 2. Specifically, GURPP leverages the expressive power of urban graphs to capture both generic and task-specific urban features. By constructing the urban region graph, GURPP effectively extracts the common spatial structures and functional layouts inherent to urban areas. Simultaneously, GURPP addresses the specificity required for different downstream tasks through a two-stage process. First, a subgraph-centric pre-training model, enhanced by multi-view self-supervised learning, captures diverse urban patterns—ranging from mobility trends and region imagery to intricate spatial and functional relationships. Then, two complementary graph-based prompting strategies further refine these representations: a manually-designed prompt that incorporates explicit task-specific semantics by emphasizing critical relationships and entities, and a task-learnable prompt that adapts to the hidden complexities of urban data. This design aims to bridge the gap between general urban knowledge and task-specific requirements for urban region representation. Contributions are as follows.

- **Graph-based Urban Region Framework:** We introduce GURPP, a novel graph-based pre-training & prompting framework for urban region representation, which unifies the modeling of complex spatial entity semantics. This design enables the extraction of heterogeneous and generic urban features, overcoming the limitations of methods that rely on predefined aggregations.
- **Subgraph-centric Pre-training and Dual Prompting Models:** We develop a novel subgraph-centric pre-training model, integrated with multi-view self-supervised learning, to capture urban generic knowledge from various modalities. Additionally, we propose two graph-based prompting methods—a manually-designed prompt and a task-learnable prompt—to inject explicit and implicit task-specific signals into the representations, thereby enhancing adaptability to diverse urban prediction tasks.
- **Extensive Empirical Validation:** We demonstrate the effectiveness of GURPP through extensive experiments on six urban region prediction tasks across two cities. Our results show significant performance improvements, with an average increase of

13.57% on MAE and 16.24% on RMSE, highlighting the model’s strong capabilities in real-world urban prediction tasks.

2 GURPP Framework

2.1 Preliminaries

Urban region. A city is divided into a set \mathcal{R} of non-overlapping regions based on defined criteria, such as administrative boundaries [30, 35, 50, 64] and grid partitioning [11, 21, 28, 62]. Each region $r \in \mathcal{R}$ consists of multiple spatial entities, including points of interest (POIs), roads, and junctions.

Urban region representation learning. The urban region representation learning problem involves learning a mapping function $\zeta : \mathcal{R} \rightarrow \mathbb{R}^d$, which generates a low-dimensional embedding $\vec{h} \in \mathbb{R}^d$ for each region $r \in \mathcal{R}$. This mapping considers both the spatial structures and functional layouts of the spatial entities within r , with d being the size of the embedding.

Once the mapping function $\zeta(\cdot)$ has been learned, it can be applied to various downstream prediction tasks, such as crime and check-in predictions. Let the task dataset be $\{(r_i, y_i)\}_{i=1}^{|\mathcal{R}|}$, where $y_i \in \mathbb{R}$ is a numerical indicator for region r_i . The downstream prediction tasks aim to predict y_i based on the region embedding $\zeta(r_i)$ using a simple regression model like a Ridge Regression [16]. However, learning effective region representations for specific tasks is challenging. First, task datasets are often very limited. This means the function $\zeta(\cdot)$ must work well in few-shot or even zero-shot situations. Second, urban spatial entities are diverse and large in number. This requires $\zeta(\cdot)$ to pull out general knowledge while still focusing on features specific to the task.

2.2 Graph-based pre-training and prompting

We propose GURPP, a unified framework that integrates urban graph pre-training with prompt learning. We formalize the GURPP framework in Eq. (1), where GURPP organizes spatial entities and inter-regional relationships within regions using a graph construction function $G(\cdot)$: During the pre-training stage \mathcal{M}_{pre}^G , the graph structure enables fine-grained feature extraction, such as capturing granular details like store brands within regions; In the prompting phase \mathcal{M}_{pmt}^G , GURPP dynamically assembles spatial entities and relationships for a region r via graph prompts \mathcal{P} to generate task-specific representations.

$$\zeta(\mathcal{P}, r; \mathcal{R}) = \mathcal{M}_{pre}^G \left(\mathcal{M}_{pmt}^G (\mathcal{P}, G(r)), G(\mathcal{R}) \right) \quad (1)$$

GURPP integrates generalizable and task-specific knowledge within a unified framework. Its effectiveness can be demonstrated by recent advancements in graph prompt learning [23, 49, 60]. Moreover, GURPP offers flexibility by supporting versatile forms of prompting for the same pre-trained model, such as manually-designed prompting (e.g., graph patterns [7, 19] and graph rules [8, 9]) or task-learnable prompting (e.g., soft prompting [5, 32]), as long as the prompt operation conforms to the urban graph schema.

In the following sections, we detail the GURPP framework with a four-step workflow in Figure 3. First, an urban graph is built to unify heterogeneous spatial entities (Section 3.1). Second, region-specific subgraphs are extracted using structural and interaction-based patterns to identify region-specific patterns and relationships

within urban data (Section 3.2). Third, a pre-trained multi-view learning model refines these subgraphs into generalizable embeddings (Section 3.3). Finally, task-specific knowledge is injected via prompt-tuning strategies (Section 4), enabling efficient adaptation without compromising the pre-trained backbone.

3 Pre-train General Representation

In this section, we introduce the processes of constructing the urban region graph and pre-training subgraph-centric region embeddings using a multi-view learning approach.

3.1 Urban Region Graph Construction

An urban region graph models the spatial structures and functional layouts of urban regions. Given a set of regions \mathcal{R} , the urban region graph, denoted as $G(\mathcal{R}) = (V, E, T_V, T_E)$ (we simplify $G(\mathcal{R})$ to G in the following), is a heterogeneous graph that captures multiple types of nodes and relationships. Here, $v \in V$ and $e \in E$ represent a node and an edge, respectively, while $\phi(v) \in T_V$ and $\psi(e) \in T_E$ are type functions that assign a specific type to a node or edge.

The urban region graph follows a schema that defines the relationships among various types of nodes and edges (see in Appendix A.1). Without loss of generality, we consider three types of spatial entities: POIs, roads, and junctions. Each spatial entity carries rich semantics; for instance, POIs have categories like shops and parks, and brands such as Dunkin’ and Starbucks. Roads include types such as highways and streets and junctions cover categories like roundabouts. This design effectively captures spatial structures and functional layouts at a granular level. For instance, two nearby regions containing coffee shops with brands of Blue Bottle and Dunkin’ are treated the same in existing statistic-based methods due to identical POI categories. In contrast, our region graph leverages semantic relations like BrandOf to distinguish them, enabling fine-grain region understanding, such as identifying high-end vs. casual commercial areas.

We construct the urban region graph from POI datasets and online maps. Since the graph contains heterogeneous nodes and edges, we first adopt the knowledge graph embedding technique TransR [27] to initialize the representations of all nodes and edges, aligning the heterogeneous information into a unified vector space.

3.2 Subgraph-centric Region Embedding

We introduce a subgraph-centric region embedding model to encode the spatial structures and functional layouts of a region r .

3.2.1 Extract region-induced subgraph. The region-induced subgraph is a subgraph of G that includes the region node r and matches the graph pattern $\mathcal{P} = (P_V, P_E, \theta)$. Here, $P_V \subseteq T_V$ and $P_E \subseteq T_E$ are sets of node and edge types, respectively, which filter entities and relationships specifically useful for region embeddings. The set $\theta \subseteq T_V$ indicates the node types that serve as termination points for subgraph extraction. As a result, the schema of the region-induced subgraph is a subset of the schema of the urban region graph. During the pre-training phase, we select all node and edge types and set $P_V = T_V, P_E = T_E$, i.e., $\mathcal{P} = (T_V, T_E, \theta)$ with the aim of comprehensively characterising the rich information of a region and thereby learning general knowledge. The region-induced subgraph $g_r = (V_r, E_r, P_V, P_E)$ includes nodes V_r and edges E_r that can be

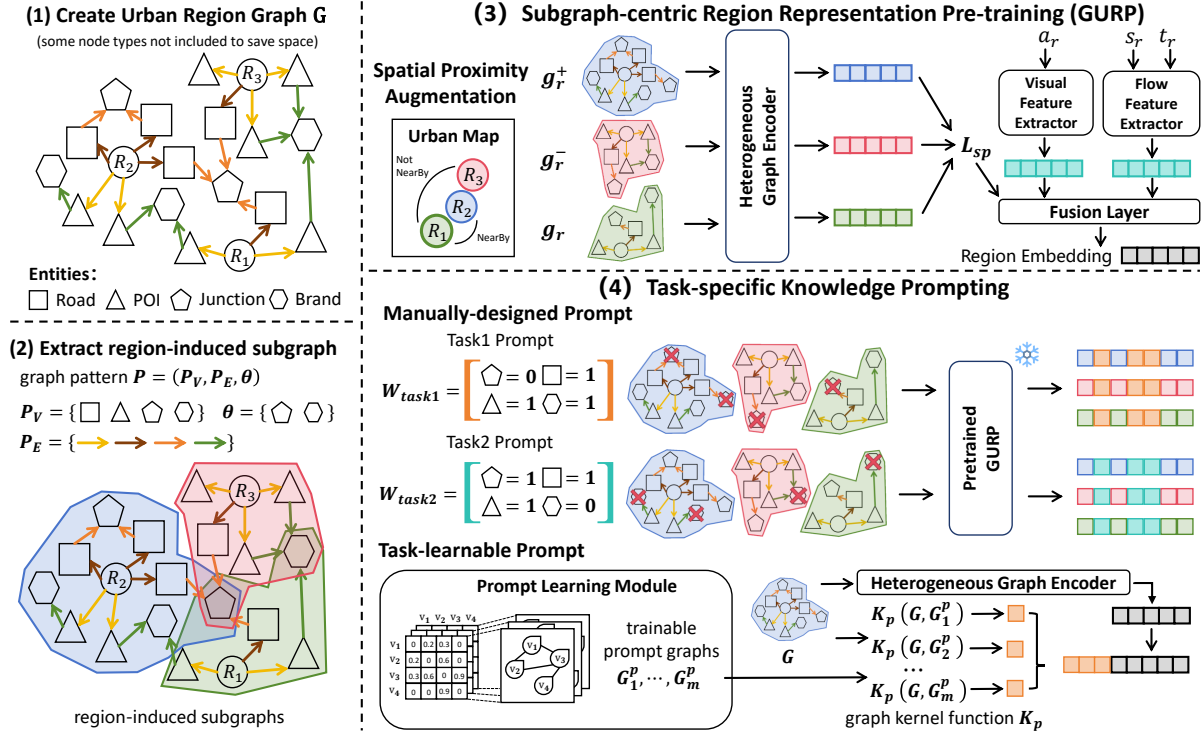


Figure 3: Overview of GURPP framework. (1) Construct the urban region graph that integrates specific urban entities and semantic relationships. (2) Extract a subgraph of each region according to the graph pattern. (3) Learn the subgraph-enhanced region representations with multi-view fusion. (4) Adapt the region embeddings with a manually-designed prompt method and a task-learnable prompt learning method.

reached from r and whose types match P_V and P_E . We extract g_r recursively, starting with $V_r = \{r\}$ and $E_r = \emptyset$ and repeatedly expanding V_r and E_r until the termination points are reached. Here, the maximum boundary for subgraph extraction is set to the two-hop neighbors of the region node according to the graph schema.

3.2.2 Region subgraph encoding. In Section 3.1, each node in the subgraph g_r has been assigned a global initial feature vector \vec{v} using TransR. We then encode each subgraph g_r into a vector \vec{h}_r to further incorporate the local features of region r . Specifically, we use a heterogeneous graph encoder $\text{gEnc}(\cdot)$ to extract the heterogeneous and generic patterns within g_r , ultimately outputting the contextualized embedding of the subgraph.

We choose the Heterogeneous Graph Transformer (HGT) [18] as the subgraph node encoder $\text{gEnc}(\cdot)$ due to its capability of capturing node and edge-type semantics. By using HGT, $\text{gEnc}(\cdot)$ characterizes the heterogeneity of input graph by maintaining a set of node and edge-type dependent parameters, and the output embedding of each node \vec{v}^{HGT} that integrates the global initial embeddings generated by TransR, and node and edge-type semantics of the subgraph g_r . This trait aligns with our need to capture the semantic features of heterogeneous entities and relationships within each region-induced subgraph.

Obtaining the representations of each node in the subgraph, we further derive the graph-level representation of g_r through two steps of aggregation. For each subgraph g_r , we first use $\text{SUM}(\cdot)$ function to aggregate the representations of nodes of the same type to

integrate the impact of the number of entities on the representation, obtaining the node-type embedding $\vec{\tau}_i$: $\vec{\tau}_i = \sum_{\phi(v)=\tau_i} \vec{v}^{HGT}$, $\tau_i \in T_V, \forall v \in V_r$. Then we concatenate all representations of node types, and encode them with a Linear layer to capture the interaction between various types of nodes: $\vec{h}_r = \text{Linear}(\text{CONCAT}_{|T_V|}(\vec{\tau}_i))$.

The output \vec{h}_r of $\text{gEnc}(\cdot)$ is the final embedding of region-induced subgraph g_r . Once the graph encoder is pretrained, any graph that conforms to the pattern P can be encoded by $\text{gEnc}(\cdot)$ to generate the corresponding representations, demonstrating high flexibility.

3.3 Multi-view Self-supervised Learning

We learn region representations in a self-supervised manner by integrating multiple views such as spatial proximity, region imagery and mobility patterns to characterize general region features.

3.3.1 Spatial proximity. The First Law of Geography [46] emphasizes the concept of spatial proximity, stating that “everything is related to everything else, but near things are more related than distant things”, which means two regions with spatial proximity should exhibit greater similarity in their representation.

We utilize a triplet network [17] to capture region similarity based on spatial proximity. For a given subgraph g_r of region r , which serves as the anchor sample, we construct its positive and negative samples based on spatial proximity. If region r' is adjacent to r (i.e., there is a “NearBy” type of edge connecting the corresponding nodes), we take the subgraph of r' as the positive sample,

denoted by g_r^+ . Otherwise, we take the subgraph as the negative sample, denoted by g_r^- . We input g_r , g_r^+ , and g_r^- to the graph encoder $\text{gEnc}(\cdot)$, obtaining the vector representations \vec{h}_r , \vec{h}_r^+ , and $\vec{h}_r^- \in \mathbb{R}^d$, respectively. Then we use triplet loss to minimize the distance between adjacent regions and maximize it between distant regions, where δ represents the margin parameter of triplet loss:

$$\mathcal{L}_{sp} = \sum_{r \in \mathcal{R}} \max\{\|\vec{h}_r - \vec{h}_r^+\|_2 - \|\vec{h}_r - \vec{h}_r^-\|_2 + \delta, 0\}.$$

3.3.2 Region imagery. Urban imagery encompasses various types of image information that describe the visual characteristics of a city, such as satellite images, street view images, and more. In this paper, we use satellite images as the source of urban imagery. For a given region r , its imagery can be represented as the set $\{a_1, a_2, \dots, a_m\}$. We employ a pre-trained visual model to extract visual features of the region \vec{h}_r^{img} and fuse these features with the region's representation with the contrastive loss \mathcal{L}_{img} (see Appendix A.2 for details).

3.3.3 Mobility pattern. Mobility pattern describes the dynamics of urban systems and is typically represented by trajectory data. For a given region r , the outflow feature s_r and inflow feature t_r are defined by calculating the number of outgoing and incoming trips within each time period. This results in the outflow and inflow feature sets $\{s_1, s_2, \dots, s_N\}$ and $\{t_1, t_2, \dots, t_N\}$, respectively, where $s_i, t_i \in \mathbb{R}^l$, and l is the number of intervals (e.g., 24). We encode the inflow and outflow features with an encoder (such as a Multi-Layer Perceptron) to obtain d -dimensional embeddings: $\vec{h}_r^{src} = \text{Encoder}(s_r)$ and $\vec{h}_r^{dst} = \text{Encoder}(t_r)$. Here we adopt the reconstruction loss \mathcal{L}_{flow} to optimize the two embeddings (see Appendix A.3 for details).

3.3.4 Self-supervised region representation learning. We aim to jointly learn region representations from spatial proximity, imagery, and flow views to extract general urban region knowledge and embed this into a unified representation space. To achieve this, we propose a view-fusion task that encodes the region embedding \vec{h}_r , the imagery embedding \vec{h}_r^{img} , and the flow embeddings \vec{h}_r^{src} and \vec{h}_r^{dst} into a combined vector \hat{h}_r . We then decode \hat{h}_r using separate decoders to reconstruct the original embeddings.

To implement this idea, we employ a fusion layer to integrate the embeddings: $\hat{h}_r = \text{ReLU}(\mathbf{W}[\vec{h}_r, \vec{h}_r^{img}, \vec{h}_r^{src}, \vec{h}_r^{dst}] + \mathbf{b})$, where $[\cdot]$ denotes the concatenation operation, and \mathbf{W} and \mathbf{b} are trainable weights. We design a multi-view prediction task to train the fusion layer with the loss function \mathcal{L}_{fuse} , defined by the ability to reconstruct the representation of each individual embedding $\vec{h}_r, \vec{h}_r^{img}, \vec{h}_r^{src}, \vec{h}_r^{dst}$ from the fused representation \hat{h}_r :

$$\mathcal{L}_{fuse} = \sum_{r \in \mathcal{R}} \sum_{k=1}^4 \|\vec{h}_r^k - \text{Decoder}^k(\hat{h}_r)\|_2^2.$$

Here, \vec{h}_r^k ($k = 1, 2, 3, 4$) represents one of the four embeddings, and Decoder^k is the corresponding decoder (implemented by a linear layer). This fusion module is easy to be extended to more views. The final loss function can be formulated as follows:

$$\mathcal{L} = \mathcal{L}_{sp} + \mathcal{L}_{img} + \mathcal{L}_{flow} + \mu \mathcal{L}_{fuse}.$$

where μ is a parameter that balances the weight of the fusion loss.

4 Prompt with Task-specific Knowledge

We propose two graph-based prompt tuning methods, a manually-designed prompt method and a task-learnable prompt learning method, to extract explicit and implicit task-specific knowledge.

4.1 Manually-designed Prompt

In the manually-designed prompt, we adjust the subgraph structure, i.e., defining the graph pattern (i.e., prompt) $\mathcal{P}(P_V, P_E, \theta)$ in Section 3.2, to emphasize specific relationships and entities relevant to the target task, as shown in Figure 4.

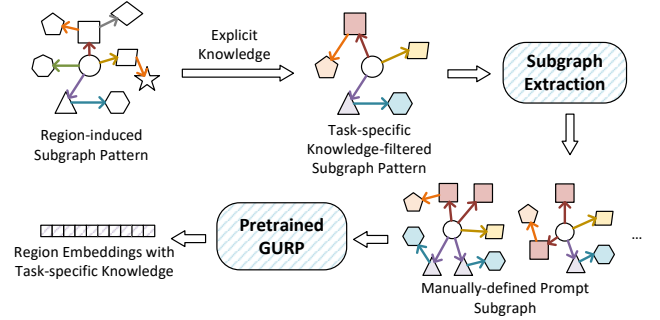


Figure 4: Manually-designed prompt.

We enhance this process by manually adjusting the proportion of different types of edges and entities in the subgraph, thereby influencing the resultant embeddings. To achieve this, we first define a set of task-specific entity weights \mathbf{W}_{task} that reflect the importance of various types of entities for the downstream task. These weights are designed based on the domain knowledge and specific requirements of task. Formally, we define $\mathbf{W}_{task} = \{w_{\tau_i} \mid \tau_i \in P_V\}$, where P_V is the set of all possible node types in region-induced subgraph, and $w_{\tau_i} \in [0, 1]$ represents the importance of entity type τ_i . The weights w_{τ_i} are manually defined and used as the deletion proportion for each entity type. We define a function $\text{Adjust}(\cdot)$ to delete nodes based on \mathbf{W}_{task} : $\text{Adjust}(g_r, \mathbf{W}_{task}) \rightarrow g'_r$, where $g_r = (V_r, E_r)$ is the original subgraph. The adjusted subgraph $g'_r = (V'_r, E'_r)$ is input to the pre-trained GURP to generate the task-specific embeddings. The function Adjust obtains the adjusted subgraph $g'_r = (V'_r, E'_r)$ as follows:

(1) Node Deletion: For each entity type $\tau_i \in \mathcal{E}$, delete $w_{\tau_i} \cdot |V_{\tau_i}|$ nodes of type τ_i from g_r , where $|V_{\tau_i}|$ is the number of nodes of type τ_i in g_r , such that $V'_r = V_r \setminus \{v \mid \phi(v) = \tau_i \text{ with probability } 1 - w_{\tau_i}\}$.

(2) Edge Deletion: After node deletion, delete edges connected to the removed nodes, such that $E'_r = \{e \mid e = (u, v) \in E; u, v \in V'_r\}$.

The manually-designed prompt is the graph pattern \mathcal{P} with the set of task-specific entity weights \mathbf{W}_{task} , which guide the $\text{Adjust}(\cdot)$ function in modifying the subgraph g_r to g'_r . Incorporating such prompts, we can tailor the urban region representations to better capture task-relevant entities and relationships.

4.2 Task-learnable Prompt

Apart from leveraging explicitly designed task-specific knowledge prompts to enrich region representations, we can also learn prompts

based on the task data and pre-trained representations to induce implicit knowledge within the task data, thereby enhancing region representations.

Considering that we describe the attributes of an urban region through subgraph induced with region nodes from G , it is intuitive to describe the task adaptation to different region attributes with graphs as well. Therefore, we propose a prompting module to use prompt graphs to portray the adaptation of downstream task relative to the original region embedding, as shown in Figure 5.

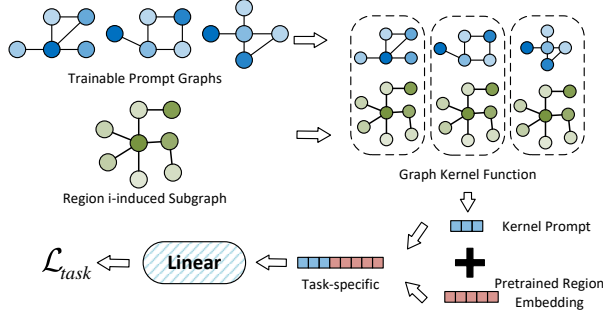


Figure 5: Task-learnable prompt.

We introduce the graph kernel method [10, 47] to measure the similarity between the prompt graphs and the region subgraphs due to its performance and computational efficiency. Let \mathcal{P} denote the set (*i.e.*, prompt) consisting of m trainable (hidden) prompt graphs, where each graph $p \in \mathcal{P}$ is defined as $p = (V^p, E^p, X^p)$. Here, $X^p \in \mathbb{R}^{|V^p| \times d}$ denotes the node attribute matrix, where d is the attribute dimension. For each region subgraph g_r , we utilize an attribute graph kernel function $K_p(\cdot, \cdot)$ to measure the similarity between g_r and each prompt graph p_1, p_2, \dots, p_m . Specifically, we employ the P -step random walk kernel (more details in Appendix A.4) to compute the kernel value between region subgraph g_r and each prompt graph p_i . These values are concatenated to form an m -dimensional prompt vector $\vec{h}_p \in \mathbb{R}^m$ for each region.

The prompt vector is used as a data operator to modify the pre-trained graph model. Conceptually, we can concatenate \vec{h}_p with each initial node feature vector \vec{v} and feed the result into a linear layer $\text{Linear}(\cdot)$, such that each entity is modified by $\vec{v}_* = \text{Linear}(\vec{h}_p || \vec{v})$. To reduce the complexity, in practice, we concatenate \vec{h}_p with the pre-trained region embedding \vec{h}_r instead of each entity embedding. To learn the prompting model, we update the prompt graphs and linear layer based on specific downstream task with a simple MSE loss function.

5 Experiments

We conduct extensive experiments on six downstream tasks of two cities from the following perspectives: (1) overall performance comparison with SOTA baselines, (2) effectiveness analysis of two prompting methods, (3) few-shot and zero-shot generalization capabilities, (4) ablation studies, (5) scalability evaluation, and (6) comparison with language-based prompting methods.

5.1 Experimental Setup

5.1.1 Datasets. We collect the datasets of region division, POI, urban imagery, taxi trips, road network and six downstream tasks for two representative cities in the United States: New York City (NYC) and Chicago (CHI).

5.1.2 Baselines. We compare our model with the following representative baselines: knowledge graph embedding methods and its variants (**TransR** [27], **TransR-N**, **TransR-G**), graph embedding methods (**node2vec** [14], **GAE** [22]), and urban region embedding methods (**MVURE** [64], **MGFN** [50], **HREP** [69], **ReCP** [25]). We follow the descriptions in these works to preprocess the data into the corresponding format and run the baselines with the parameter settings specified in the papers to ensure fairness.

5.1.3 Metrics and Implementation. We use three widely recognized evaluation metrics to evaluate the prediction performance: the coefficient of determination (R^2), root mean squared error (RMSE), and mean absolute error (MAE). Better performance is indicated by a higher R^2 and lower RMSE and MAE values. Our model includes three version: 1) **GURP** represents the pre-train module, 2) **GURPP_M** is GURP with manually-designed prompt, and 3) **GURPP_T** is GURP with task-learnable prompt. Details on datasets and implementation are in Appendix A.5.

5.2 Overall Performance

We evaluate the overall performance of the baselines and our model on datasets collected from two cities, covering six different downstream tasks. The results of the experiments are shown in Table 1, from which we find the following.

(1) GURPP demonstrates superior performance across both cities and tasks. In the pre-training phase, GURP achieves state-of-the-art performance on 6 out of 8 tasks, with particularly significant improvements on CHI datasets (27.03%-31.95% MAE reduction on Crime/Crash prediction). When enhanced with prompting methods, GURPP further improves performance by an average of 11.30% (MAE) and 10.16% (RMSE) across all tasks compared to GURP. The consistent improvements across heterogeneous tasks (12.46% average MAE gain and 14.62% RMSE gain over baselines) validate the effectiveness of our pre-training and prompting framework.

(2) The results highlight our model's exceptional adaptability to different urban contexts. Specifically, GURPP achieves average improvements of 24.2% (MAE) and 25.9% (RMSE) in CHI versus 3.0% (MAE) and 6.6% (RMSE) in NYC. This may be due to the fact that the baselines tend to be effective only at extracting information specific to certain data types, but they do not consider spatial structure and functional layouts among urban entities. For example, existing baselines rely on the flow view for information extraction, but the sparse content in the CHI dataset significantly degrades downstream task performance. In contrast, GURPP performs excellently on both urban datasets, demonstrating its robustness and universality across diverse urban environments.

(3) To further validate the effectiveness of our subgraph-centric and multi-view fusion module, we conduct experiments comparing the performance of models with and without these modules. Specifically, the result that TransR-G outperforms TransR-N by an average of 1.27% (MAE) indicates that the subgraph-centric method

Table 1: Main performance comparison of different models on downstream tasks. The best results for each metric are highlighted in bold, the second-best are underlined, and the third-best are marked with an asterisk.

Model	New York City (NYC)								Chicago (CHI)							
	Crime		Check-in		Carbon		Income		Crime		Crash		Carbon		Education	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
TransR-N	111.32	148.86	534.70	805.28	115.65	175.73	36799	45098	209.07	307.45	106.91	176.86	97.15	126.59	653.68	932.56
TransR-G	109.22	146.91	530.43	802.43	116.80	179.25	36363	44571	208.40	306.16	104.68	174.42	95.76*	124.37*	601.97	875.24
node2vec	138.46	175.80	573.12	837.44	174.45	225.42	45330	56182	246.07	349.97	121.90	174.91	115.62	148.95	832.89	1139.33
GAE	96.85	126.36	349.94	542.92	115.99	174.35	35338	43200	200.61*	300.24	100.54	173.32	<u>95.29</u>	<u>123.66</u>	702.82	986.70
MVURE	72.47*	99.84	310.97	498.08	<u>99.30</u>	<u>151.01</u>	24031	35351	206.08	289.86*	96.95	148.87*	98.19	130.31	593.80	843.10
MGFN	73.50	97.44*	300.15	456.63*	136.49	185.82	30266	40717	214.02	310.83	108.37	179.85	96.56	125.50	709.91	998.03
HRE	76.93	102.07	329.55	513.80	102.95	160.58	25903	35418	206.61	298.39	99.95	158.63	97.11	133.00	555.56	806.60
HREP	74.85	100.04	320.34	505.52	102.49	159.68	24643	<u>34148</u>	206.59	298.24	99.77	158.66	97.19	132.91	555.01	806.36
ReCP	73.85	100.56	292.87*	524.89	95.55	150.84	24440*	35044	207.49	298.97	93.30*	157.52	95.84	125.73	455.98*	708.37*
GURP	72.09	93.22	271.69	412.92	108.17	158.03	25268	34943*	161.27	225.24	80.36	<u>108.78</u>	97.83	133.03	455.43	677.36
GURPP	71.48	91.61	261.45	388.23	100.53*	152.70*	22902	31945	146.39	203.05	65.30	92.11	64.844	88.56	420.62	656.98
Enhance	1.36%	5.98%	10.73%	14.98%	-4.95%	-1.21%	4.70%	6.45%	27.03%	29.95%	30.01%	38.12%	31.95%	28.39%	7.76%	7.26%

can preserve the structural and contextual information of regions, providing more effective urban region embeddings. The average 32.55% (MAE) improvement for GURP over TransR-G is mainly due to the heterogeneous feature extraction, which improves the model’s comprehensive understanding of urban features.

5.3 Effectiveness of Prompting Methods

5.3.1 Manually-designed Prompt Study. In the real world, crime rates are closely linked to specific POI categories, roads, and junctions, with locations like ATMs, bars, and parks being more crime-prone. Road density and junction complexity also impact crime. Regarding user check-ins, brand popularity plays a key role, with users often preferring well-known brands like Starbucks or McDonald’s. Based on these observations, we design two patterns: P1 (ignoring POI brand information) and P2 (randomly sampling POI, road, and junction nodes with a 0.9 ratio) to evaluate the performance of manually-designed prompts across different tasks.

Table 2: Performance of manually-designed prompt GURPP_M.

Model	Crime Prediction (NYC)			Check-in Prediction (NYC)		
	MAE	RMSE	R ²	MAE	RMSE	R ²
GURP	72.094	93.216	0.603	<u>271.689</u>	<u>412.918</u>	<u>0.745</u>
GURPP _M -P1	68.538	88.698	0.640	273.201	429.564	0.724
GURPP _M -P2	<u>69.421</u>	<u>91.834</u>	<u>0.614</u>	262.437	397.466	0.764

Results in Table 2 show that focusing on factors such as POI categories, roads, and junctions, closely related to criminal behavior, can improve crime prediction performance. Meanwhile, removing POI brand information shows a significant decrease in the performance of check-in prediction. This demonstrates that our manually-designed prompting method can improve the prediction performance of downstream tasks with explicit knowledge injection.

5.3.2 Task-learnable Prompt Study. We utilize GURP pre-trained embedding with the simple prefix prompt (proposed by HREP[69] denoted as **GURP+Pre**), which directly concatenates hidden vector before pretrained embedding without modeling task-adaptation, and our task-learnable prompt for comparative analysis. Both of these prompt methods are data-driven and seek to improve the

Table 3: Performance of task-learnable prompt GURPP_T.

Framework	Crime Prediction (NYC)			Crash Prediction (CHI)		
	MAE	RMSE	R ²	MAE	RMSE	R ²
GURP	<u>72.094</u>	<u>93.216</u>	<u>0.603</u>	80.362	108.781	0.622
GURP+Pre	73.758	93.971	0.596	<u>66.330</u>	<u>93.772</u>	<u>0.719</u>
GURPP _T	71.481	91.612	0.616	65.297	92.114	0.729

region representation for specific downstream tasks. We conduct experiments on crime prediction task in NYC and crash prediction task in CHI, respectively, and the results are shown in Table 3.

Results show that our task-learnable prompt outperforms pre-trained embeddings across both tasks, with improvements of 2.16% (R^2) in crime prediction and 17.20% (R^2) in crash prediction, indicating that the prompt is highly effective in providing tailored guidelines for different downstream tasks. In addition, GURPP_T outperforms the prefix prompt in both tasks. This is because the prefix prompt only utilizes the pre-trained region representations for prompting, while our graph-based prompting method also incorporates the original region subgraphs to prompt, better capturing the correlations between task and region attributes.

5.3.3 Comparison of Two Prompting Methods. We compare the performance of two proposed prompting methods across different cities and tasks to evaluate their effectiveness in various scenarios. In addition to the two prompting patterns P1 and P2, we also define two additional patterns: P3 (randomly samples POI with a weight of 0.9 and ignores road and junction category) and P4 (randomly samples POI with a weight of 0.7 and ignores road and junction), to further validate the effectiveness of different prompting methods. Results show in Table 4.

From the results, we observe that when the pre-trained city region data is relatively sparse (e.g., in cases of CHI), task-learnable prompting methods, especially our GURPP_T, achieve better enhancement effects. Conversely, when the data is more abundant (e.g., in cases of NYC), manually-designed prompts with explicitly injected knowledge perform better. Moreover, different manually-designed prompts explicitly capture distinct types of knowledge, leading to varying performance across different tasks, which demonstrates the flexibility of our proposed graph-based prompting approach.

Table 4: Comparison of two prompting methods.

Method	New York City (NYC)			Chicago (CHI)		
	Check-in	Carbon	Income	Crime	Carbon	Education
	RMSE	R^2	MAE	RMSE	R^2	MAE
GURP	412.92	<u>0.23</u>	25268	<u>225.24</u>	<u>-0.33</u>	<u>455.43</u>
GURPP _T	<u>388.23</u>	0.17	<u>24745</u>	203.05	0.41	420.62
GURPP _M -P1	429.56	-0.13	27839	234.26	-0.35	465.81
GURPP _M -P2	397.47	-0.22	26392	234.74	-0.40	458.59
GURPP _M -P3	464.90	0.28	28170	256.53	-0.66	486.19
GURPP _M -P4	340.25	0.16	22902	238.36	-0.49	502.86

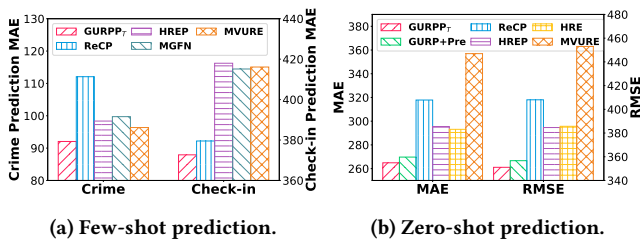
5.4 Few-shot & Zero-shot Performance

We compare the performance of GURPP_T and baselines in few-shot and zero-shot scenarios to validate the generalizability of pre-trained representations and superiority of prompting methods.

5.4.1 Few-shot Prediction. With the pre-trained region representations as input, we randomly sample 10% of the region data as the training set to develop the prediction model, while the remaining 90% is designated as the testing set, to evaluate the performance of GURPP_T with other baselines in few-shot scenarios.

Figure 6a shows that GURPP_T outperforms all baselines in the few-shot scenario. This is partly because the pre-trained representations extract generic regional features that exhibit better adaptability in low-resource settings, which, when combined with task-learnable prompts, can better accommodate few-shot prediction. Additionally, our prompting method shows effective transferability compared to HREP.

5.4.2 Zero-shot Prediction. In this experiment, we first trained the prediction model on NYC crime datasets with NYC pre-trained region embeddings, and then evaluated it on CHI crime datasets without any fine-tuning. This setup enables a comparative analysis of the zero-shot transferability between our task-learnable prompt and various baselines. Results show in Figure 6b.

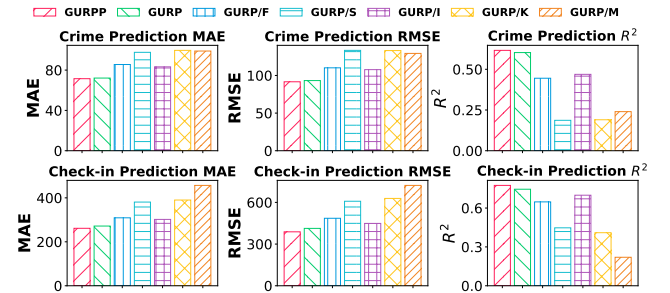
**Figure 6: Few-shot & zero-shot performance.**

Compared with baselines without any prompting methods (i.e., HRE, MVURE, and ReCP), models with prompting generally exhibit better zero-shot transferability, suggesting that prompting model can capture task-specific features independent of the city to some extent. Under the same prompting method (prefix prompt), GURP+Pre demonstrates better transferability than HREP, highlighting the superior generalizability of our GURP pre-trained representations. A comparison between GURPP and GURP+Pre reveals that our task-learnable prompting model outperforms the prefix prompt, which can be attributed to the handling of the original region subgraphs within our prompting model.

5.5 Ablation Studies

We perform ablation studies on the NYC dataset to evaluate the effectiveness of each individual component of our proposed model. GURPP includes both pretraining and task-learnable prompting module. GURP contains only the pretraining module. /F, /S, /I, /M denote the removal of the flow view, spatial proximity view, imagery view, and multi-view fusion from GURP, respectively. /K indicates replacing the knowledge graph embedding initialization with random initialization.

Figure 7 presents the results. Spatial proximity view and knowledge graph embedding method significantly enhance the model's performance, lifting it by 151.64% and 156.67% of R^2 , respectively. These components capture the heterogeneous and generic patterns of interactions among entities by elucidating the spatial relationships and semantic knowledge of urban regions. The removal of the multi-view fusion module results in learned representations exerting varying effects across different tasks, which indicates that this module contributes to learning more stable representations. The flow view and the imagery view can also greatly enhance the performance of the model, increasing it by 29.04% and 21.11%, respectively. This is because they capture dynamic flow information and visual features of urban regions, respectively, providing the model with a multi-dimensional context that enhances its ability to fully understand and predict urban dynamics. Prompt learning, with an improvement of 3.12%, effectively elevates model performance by introducing guiding prompts that allow the model to delve deeper into understanding and utilizing task-relevant knowledge.

**Figure 7: Ablation studies.**

5.6 Scalability Analysis

We also validate the scalability of our graph-based method by measuring the pre-training and prompting tuning duration across various scales of regions and region subgraph nodes. Specifically, we use 180 regions in NYC as the base and evaluate the training duration of GURPP over 500 epochs and prompt tuning over 200 epochs for region scale factors of 0.1, 0.2, up to 5 times (1.6 times for pre-training), and evaluate the pretraining duration using 0.05, 0.1, up to 1.0 times the total number of nodes in each region subgraph. For prompt tuning, we conduct evaluations with subgraphs containing 1, 5, up to 120 nodes with the base number 50, which is also the actual experiments settings. The results are averaged over each epoch, shown in Figure 8.

As shown in Figure 8a, when the region subgraph size is fixed (GURP uses all nodes and prompt tuning uses 50 nodes per subgraph), the training time for pretraining exhibits a superlinear

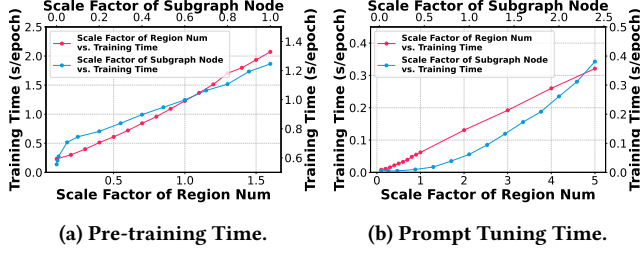


Figure 8: Impact of data scale on training time.

growth with the number of regions. This is mainly due to the flow feature extraction, where the model incorporates dynamics between all region pairs, whereas prompt tuning scales linearly with the number of regions. When the number of regions is fixed, the pretraining duration shows a sublinear relationship with the number of nodes in the regional subgraph, ensuring the scalability of GURP. Due to the random walk kernel, the training time for the task-learnable prompt scales superlinearly with the number of nodes. We set the number of nodes to 50, maintaining acceptable training efficiency.

5.7 Case Study

To validate the superiority of GURPP, we conduct a case study to predict the top 10 regions with the highest crime rates in NYC. The study compares our manually designed prompt, GURPP_M-P4, against large language models (LLMs), including Deepseek R1 (www.deepseek.com) and Gemini 2 Flash (deepmind.google/technologies/gemini/flash/). Detailed descriptions are in Appendix A.5.4.

As shown in the Figure 9, GURPP is able to predict crime regions (the regions in the upper half of the figure) but LLMs fail. This indicates that, compared to language prompts, which directly describe the task to induce relevant knowledge in large language models, GURPP explicitly incorporates task-specific knowledge into the representation using graphs, leading to better performance.

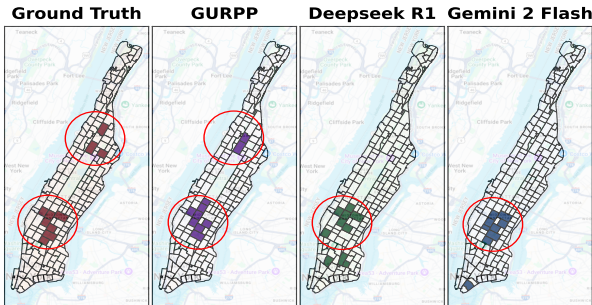


Figure 9: Comparison with language prompt. Shaded regions represent potential crime regions.

6 Related Work

Urban Region Embedding. Urban region representation learning aims to extract complex patterns and relationships within urban regions and embed them into a representation space. Existing research can be broadly classified into two categories: task-specific supervised region representation learning [2, 55, 70] and general unsupervised region representation learning [20, 24, 25, 42, 50, 53, 57, 63, 64]. These works, focusing more on exploring multi-view fusion methods to generate representation based on predefined region statistics

features with limitations in capturing detailed spatial layout and semantics. Moreover, the lack of fine-grained characterization of region features makes it challenging to further adapt the representation to the specificities of downstream tasks. We address the limitations by using an urban region graph and subgraph-centric region embeddings. Although Zhao *et al.* [66] employ a graph-based approach to model urban regions at a fine-grained level, their framework is not specifically designed for urban-region pre-training and prompting tasks.

Pre-training and Prompting. The Pre-training and prompting paradigms have achieved significant success in NLP [29, 39], CV [58, 67, 68], and graph learning [33, 34, 43, 45, 56, 59, 61]. Many works in urban computing have also made attempts in this regard [3, 26, 51, 52, 54, 62, 65]. For instance, Yuan *et al.* [62] design an urban spatio-temporal prediction pre-train and prompt model. Yan *et al.* [54] introduce LLMs to describe urban imagery features and unify different prediction tasks through textual prompts. Although the urban tasks they explored differ from ours, their work can inspire ours to integrate LLMs to enhance fine-grained characterization capabilities. HREP [69] introduces prefix prompt to region embedding, but it lacks explicit extraction of task-adaptation. In this way, we propose both manually-designed and task-learnable prompts built on the graph-based pre-train model to address this issue. The categorization of related work is listed in Table 5.

Table 5: Cats.of related works on urban region embedding.

Category	Non-Graph-based	Graph-based
Pretraining	[24, 25, 51, 54, 63]	[13, 50, 64, 69], GURPP (Ours)
Prompting	[54, 69]	GURPP (Ours)

7 Conclusion and Future Works

In this paper, we introduce the urban region graph and propose a graph-based pre-training and prompting framework for urban region representation learning, balancing urban generality and task specificity. We develop a subgraph-centric pre-training model to capture general urban knowledge and two prompt methods to incorporate task-specific insights. Extensive experiments demonstrate the effectiveness of our approach. Future work may explore: 1) Integrating textual data and LLMs to uncover more universal urban knowledge; 2) Developing arbitrary region representations for flexible urban modeling [41]; 3) Assessing the transferability of urban representations across cities to enhance general applicability.

References

- [1] Arcgis 2024. *ArcGIS Online*. Retrieved July 28, 2024 from <https://www.arcgis.com/index.html>
- [2] Lei Bai, Lina Yao, Salil S Kanhere, Xianzhi Wang, Wei Liu, and Zheng Yang. 2019. Spatio-temporal graph convolutional and recurrent networks for citywide passenger demand prediction. In *Proceedings of the 28th ACM international conference on information and knowledge management (CIKM)*. 2293–2296.
- [3] Pasquale Balsebre, Weiming Huang, Gao Cong, and Yi Li. 2023. City Foundation Models for Learning General Purpose Representations from OpenStreetMap. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM)*. 87–97.
- [4] Census Bureau 2024. *Measuring America's People, Places, and Economy*. Retrieved July 28, 2024 from <https://www.census.gov/>
- [5] Chen Chen, Yufei Wang, Aixin Sun, Bing Li, and Kwok-Yan Lam. 2023. Dipping PLMs sauce: bridging structure and text for effective knowledge graph completion via conditional soft prompting. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*. 11489–11503.

- [6] Chicago Data Portal 2024. *Chicago Data Portal*. Retrieved July 28, 2024 from <https://data.cityofchicago.org/>
- [7] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 135–144.
- [8] Wenfei Fan, Wenzhi Fu, Ruochun Jin, Ping Lu, and Chao Tian. 2022. Discovering association rules from big graphs. *Proceedings of the VLDB Endowment* 15, 7 (2022), 1479–1492.
- [9] Wenfei Fan, Ziyang Han, Yaoshu Wang, and Min Xie. 2022. Parallel rule discovery from large datasets by sampling. In *Proceedings of the 2022 International Conference on Management of Data (SIGMOD)*. 384–398.
- [10] Aosong Feng, Chenyu You, Shiqiang Wang, and Leandros Tassioulas. 2022. Kergnns: Interpretable graph neural networks with graph kernels. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, Vol. 36. 6614–6622.
- [11] Kaiyu Feng, Gao Cong, Christian S Jensen, and Tao Guo. 2019. Finding attribute-aware similar region for data analysis. *Proceedings of the VLDB Endowment* 12, 11 (2019), 1414–1426.
- [12] Foursquare 2024. *What can Foursquare do for you?* Retrieved July 28, 2024 from <https://data.cityofchicago.org/>
- [13] Yanjie Fu, Pengyang Wang, Jiadi Du, Le Wu, and Xiaolin Li. 2019. Efficient region embedding with multi-view spatial networks: A perspective of locality-constrained spatial autocorrelations. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 33. 906–913.
- [14] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 855–864.
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778.
- [16] Arthur E Hoerl and Robert W Kennard. 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 12, 1 (1970), 55–67.
- [17] Elad Hoffer and Nir Ailon. 2015. Deep metric learning using triplet network. In *Proceedings of Similarity-based pattern recognition: third international workshop (SIMBAD)*, Springer, 84–92.
- [18] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of the Web conference 2020 (WWW)*. 2704–2710.
- [19] Kexin Huang and Marinka Zitnik. 2020. Graph meta learning via local subgraphs. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS)*, Vol. 33. 5862–5874.
- [20] Porter Jenkins, Ahmad Farag, Suhang Wang, and Zhenhui Li. 2019. Unsupervised representation learning of spatial data via multimodal embedding. In *Proceedings of the 28th ACM international conference on information and knowledge management (CIKM)*. 1993–2002.
- [21] Jiahui Jin, Yifan Song, Dong Kan, Binjie Zhang, Yan Lyu, Jinghui Zhang, and Hongru Lu. 2024. Learning context-aware region similarity with effective spatial normalization over Point-of-Interest data. *Information Processing & Management* 61, 3 (2024), 103673.
- [22] Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).
- [23] Jia Li, Xiangguo Sun, Yuhua Li, Zhixun Li, Hong Cheng, and Jeffrey Xu Yu. 2024. Graph intelligence with large language models and prompt learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 6545–6554.
- [24] Yi Li, Weiming Huang, Gao Cong, Hao Wang, and Zheng Wang. 2023. Urban region representation learning with OpenStreetMap building footprints. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 1363–1373.
- [25] Zechen Li, Weiming Huang, Kai Zhao, Min Yang, Yongshun Gong, and Meng Chen. 2024. Urban Region Embedding via Multi-View Contrastive Prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 38. 8724–8732.
- [26] Zhonghang Li, Lianghao Xia, Jiabin Tang, Yong Xu, Lei Shi, Long Xia, Dawei Yin, and Chao Huang. 2024. Urbangpt: Spatio-temporal large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 5351–5362.
- [27] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI conference on Artificial Intelligence (AAAI)*, Vol. 29. 2181–2187.
- [28] Lingbo Liu, Zhilin Qiu, Guanbin Li, Qing Wang, Wanli Ouyang, and Liang Lin. 2019. Contextualized spatial-temporal network for taxi origin-destination demand prediction. *IEEE Transactions on Intelligent Transportation Systems* 20, 10 (2019), 3875–3887.
- [29] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Comput. Surveys* 55, 9 (2023), 1–35.
- [30] Yu Liu, Jingtao Ding, Yanjie Fu, and Yong Li. 2023. Urbankg: An urban knowledge graph system. *ACM Transactions on Intelligent Systems and Technology* 14, 4 (2023), 1–25.
- [31] Yu Liu, Xin Zhang, Jingtao Ding, Yanxin Xi, and Yong Li. 2023. Knowledge-infused contrastive learning for urban imagery-based socioeconomic prediction. In *Proceedings of the ACM Web Conference 2023 (WWW)*. 4150–4160.
- [32] Zheyuan Liu, Xiaoxin He, Yijun Tian, and Nitesh V Chawla. 2024. Can we soft prompt LLMs for graph learning tasks?. In *Proceedings of the ACM Web Conference 2024 (WWW)*. 481–484.
- [33] Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. 2023. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. In *Proceedings of the ACM Web Conference 2023 (WWW)*. 417–428.
- [34] Yihong Ma, Ning Yan, Jiayu Li, Masood Mortazavi, and Nitesh V Chawla. 2024. Hetgpt: Harnessing the power of prompt tuning in pre-trained heterogeneous graph neural networks. In *Proceedings of the ACM Web Conference 2024 (WWW)*. 1015–1023.
- [35] Yansong Ning, Hao Liu, Hao Wang, Zhenyu Zeng, and Hui Xiong. 2024. UUKG: unified urban knowledge graph dataset for urban spatiotemporal prediction. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS)*, Vol. 36. 62442–62456.
- [36] NYC OpenData 2024. *Open Data for All New Yorkers*. Retrieved July 28, 2024 from <https://opendata.cityofnewyork.us/>
- [37] Tomohiro Oda and Shamil Maksyutov. 2015. ODIAC fossil fuel CO2 emissions dataset (Version name: ODIAC2023). *Center for Global Environmental Research (CGER), National Institute for Environmental Studies* (2015).
- [38] OpenStreetMap 2024. *OpenStreetMap*. Retrieved July 28, 2024 from <https://www.openstreetmap.org/>
- [39] Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How Much Knowledge Can You Pack Into the Parameters of a Language Model?. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 5418–5426.
- [40] Safegraph 2024. *Your Partner in Places Data*. Retrieved July 28, 2024 from <https://www.safegraph.com/>
- [41] Fengze Sun, Yanchuan Chang, Egemen Tanin, Shanika Karunasekera, and Jianzhong Qi. 2025. FlexiReg: Flexible Urban Region Representation Learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*.
- [42] Fengze Sun, Jianzhong Qi, Yanchuan Chang, Xiaoliang Fan, Shanika Karunasekera, and Egemen Tanin. 2024. Urban region representation learning with attentive fusion. In *Proceedings of 2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 4409–4421.
- [43] Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, and Jihong Guan. 2023. All in one: Multi-task prompting for graph neural networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 2120–2131.
- [44] Xiangguo Sun, Jiawen Zhang, Xixi Wu, Hong Cheng, Yun Xiong, and Jia Li. 2024. Graph Prompt Learning: A Comprehensive Survey and Beyond. *arXiv preprint arXiv:2311.16534* (2024).
- [45] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Long Xia, Dawei Yin, and Chao Huang. 2024. HiGPT: Heterogeneous Graph Language Model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 2842–2853.
- [46] Waldo R Tobler. 1970. A computer movie simulating urban growth in the Detroit region. *Economic geography* 46, sup1 (1970), 234–240.
- [47] S Vichy N Vishwanathan, Nicol N Schraudolph, Risi Kondor, and Karsten M Borgwardt. 2010. Graph kernels. *The Journal of Machine Learning Research* 11 (2010), 1201–1242.
- [48] Hongjian Wang and Zhenhui Li. 2017. Region representation learning via mobility flow. In *Proceedings of the 26th ACM International Conference on Information and Knowledge Management (CIKM)*. 237–246.
- [49] Qunzhong Wang, Xiangguo Sun, and Hong Cheng. 2025. Does Graph Prompt Work? A Data Operation Perspective with Theoretical Analysis. In *Proceedings of Forty-Second International Conference on Machine Learning (ICML)*.
- [50] Shangbin Wu, Xu Yan, Xiaoliang Fan, Shirui Pan, Shichao Zhu, Chuanpan Zheng, Ming Cheng, and Cheng Wang. 2022. Multi-graph fusion networks for urban region embedding. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*. 2312–2318.
- [51] Congxi Xiao, Jingbo Zhou, Yixiong Xiao, Jizhou Huang, and Hui Xiong. 2024. ReFound: Crafting a Foundation Model for Urban Region Understanding upon Language and Visual Foundations. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 3527–3538.
- [52] Fengli Xu, Jun Zhang, Chen Gao, Jie Feng, and Yong Li. 2023. Urban generative intelligence (ugi): A foundational platform for agents in embodied city environment. *arXiv preprint arXiv:2312.11813* (2023).
- [53] Zhuo Xu and Xiao Zhou. 2024. CGAP: urban region representation learning with coarsened graph attention pooling. *Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI)* (2024), 7518–7526.
- [54] Yibo Yan, Haomin Wen, Siru Zhong, Wei Chen, Haodong Chen, Qingsong Wen, Roger Zimmermann, and Yuxuan Liang. 2024. Urbanclip: Learning text-enhanced

- urban region profiling with contrastive language-image pretraining from the web. In *Proceedings of ACM Web Conference 2024 (WWW)*. 4006–4017.
- [55] Min Yang, Bo Kong, Ruirong Dang, and Xiongfeng Yan. 2022. Classifying urban functional regions by integrating buildings and points-of-interest using a stacking ensemble method. *International Journal of Applied Earth Observation and Geoinformation* 108 (2022), 102753.
- [56] Yuhao Yang, Lianghao Xia, Da Luo, Kangyi Lin, and Chao Huang. 2024. GraphPro: Graph Pre-training and Prompt Learning for Recommendation. In *Proceedings of the ACM Web Conference 2024 (WWW)*. 3690–3699.
- [57] Zijun Yao, Yanjie Fu, Bin Liu, Wangsu Hu, and Hui Xiong. 2018. Representing urban functions through zone embedding with human mobility patterns. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*. 3919–3925.
- [58] Seungryong Yoo, Eunji Kim, Dahuin Jung, Jungbeom Lee, and Sungroh Yoon. 2023. Improving visual prompt tuning for self-supervised vision transformers. In *Proceedings of Fortieth International Conference on Machine Learning (ICML)*. 40075–40092.
- [59] Xingtong Yu, Yuan Fang, Zemin Liu, and Xinming Zhang. 2024. Hgprompt: Bridging homogeneous and heterogeneous graphs for few-shot prompt learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 38. 16578–16586.
- [60] Xingtong Yu, Zhenghao Liu, Yuan Fang, Zemin Liu, Sihong Chen, and Xinming Zhang. 2024. Generalized Graph Prompt: Toward a Unification of Pre-Training and Downstream Tasks on Graphs. *IEEE Transactions on Knowledge & Data Engineering* 36, 11 (2024), 6237–6250.
- [61] Xingtong Yu, Chang Zhou, Yuan Fang, and Xinming Zhang. 2024. MultiGPrompt for multi-task pre-training and prompting on graphs. In *Proceedings of the ACM Web Conference 2024 (WWW)*. 515–526.
- [62] Yuan Yuan, Jingtao Ding, Jie Feng, Depeng Jin, and Yong Li. 2024. UniST: A Prompt-Empowered Universal Model for Urban Spatio-Temporal Prediction. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 4095–4106.
- [63] Liang Zhang, Cheng Long, and Gao Cong. 2023. Region embedding with intra and inter-view contrastive learning. *IEEE Transactions on Knowledge and Data Engineering* 35, 9 (2023), 9031–9036.
- [64] Mingyang Zhang, Tong Li, Yong Li, and Pan Hui. 2021. Multi-view joint graph representation learning for urban region embedding. In *Proceedings of the 29th International Joint Conferences on Artificial Intelligence (IJCAI)*. 4431–4437.
- [65] Weijia Zhang, Jindong Han, Zhao Xu, Hang Ni, Hao Liu, and Hui Xiong. 2024. Towards Urban General Intelligence: A Review and Outlook of Urban Foundation Models. *arXiv preprint arXiv:2402.01749* (2024).
- [66] Yunxiang Zhao, Jianzhong Qi, Bayu D Trisedya, Yixin Su, Rui Zhang, and Hongguang Ren. 2023. Learning region similarities via graph-based deep metric learning. *IEEE Transactions on Knowledge and Data Engineering* 35, 10 (2023), 10237–10250.
- [67] Kaiyang Zhou, Jingkan Yang, Chen Change Loy, and Ziwei Liu. 2022. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 16816–16825.
- [68] Kaiyang Zhou, Jingkan Yang, Chen Change Loy, and Ziwei Liu. 2022. Learning to prompt for vision-language models. *International Journal of Computer Vision* 130, 9 (2022), 2337–2348.
- [69] Silin Zhou, Dan He, Lisi Chen, Shuo Shang, and Peng Han. 2023. Heterogeneous region embedding with prompt learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 37. 4981–4989.
- [70] Zhengyang Zhou, Kuo Yang, Yuxuan Liang, Binwu Wang, Hongyang Chen, and Yang Wang. 2023. Predicting collective human mobility via countering spatiotemporal heterogeneity. *IEEE Transactions on Mobile Computing* 23, 5 (2023), 4723–4738.

A Appendix

A.1 Urban Region Graph Construction

Here we provide more details for the urban region graph construction, with the schema shown in Figure 10. We define eight node types shared across regions and cities: *region*, *POI*, *POI category*, *brand*, *road*, *road category*, *junction*, and *junction category* in total, comprising the node type set T_V , and six edge types to form the edge type set T_E . These include the *NearBy* relationship, which connects region entities based on their geographical adjacency; the *Contains* relationship, which is based on spatial containment; the *BrandOf* relationship, which links brands to POI entities; and the

CateOf, *JCateOf*, and *RCateOf* relationships, which are based on the category information of the entities.

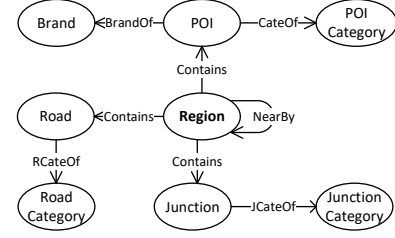


Figure 10: Schema of urban region graph.

A.2 Region Imagery Encoding

We provide details on urban imagery encoding. ResNet [15] is utilized to extract visual features for each region image, represented as $\vec{a}_k = \text{ResNet}(a_k)$. A comprehensive visual representation for the region is obtained by averaging features from multiple images: $\vec{h}_r^{img} = \text{Linear}(\frac{1}{m} \sum_{k=1}^m \vec{a}_k)$. We then adopt the contrastive loss \mathcal{L}_{img} to align the visual representation with the subgraph-centric embedding.

$$\mathcal{L}_{img} = \sum_{r \in \mathcal{R}} \left[-\log \exp \left(\frac{\vec{h}_r \cdot \vec{h}_r^{img}}{\tau} \right) + \log \left(\exp \left(\frac{\vec{h}_r \cdot \vec{h}_r^{img}}{\tau} \right) + \sum_{j=1}^{B-1} \exp \left(\frac{\vec{h}_r \cdot \vec{h}_j}{\tau} \right) \right) \right],$$

where, B represents batch size and τ is the temperature parameter.

A.3 Mobility Pattern Encoding

We provide a detailed description for mobility pattern encoding. We first reconstruct the departure and arrival flow distribution. Let m_{ij} represent the number of trips from region r_i to region r_j . The source and destination urban flow distributions are computed as $p^{sc}(r_j|r_i) = \frac{m_{ij}}{\sum_{k=1}^N m_{ik}}$ and $p^{dt}(r_i|r_j) = \frac{m_{ij}}{\sum_{k=1}^N m_{kj}}$, respectively. We then reconstruct the source and destination distributions $\hat{p}^{sc}(r_j|r_i)$ and $\hat{p}^{dt}(r_i|r_j)$ using the encoded inflow and outflow features:

$$\hat{p}^{sc}(r_j|r_i) = \frac{\exp((\vec{h}_i^{sc})^\top \vec{h}_j^{dt})}{\sum_k \exp((\vec{h}_i^{sc})^\top \vec{h}_k^{dt})}, \quad \hat{p}^{dt}(r_i|r_j) = \frac{\exp((\vec{h}_j^{dt})^\top \vec{h}_i^{sc})}{\sum_k \exp((\vec{h}_j^{dt})^\top \vec{h}_k^{sc})}.$$

Finally, we learn the inflow and outflow embeddings by optimizing the following loss function:

$$\mathcal{L}_{flow} = \sum_{(r_i, r_j)} -p^{sc}(r_j|r_i) \log \hat{p}^{sc}(r_j|r_i) - p^{dt}(r_i|r_j) \log \hat{p}^{dt}(r_i|r_j),$$

which aids in learning inflow and outflow embeddings by comparing true and reconstructed distributions.

A.4 Attribute Graph Kernel Function

We introduce how attribute graph kernel function $K_p(\cdot, \cdot)$ works. We choose a P -step random walk kernel [10] as graph kernel function $K(\cdot, \cdot)$ to measure the similarity between graphs, which counts the number of common walks in both graphs within P steps. Given $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$, where V and E are the set of nodes and edges, we initially define the direct product graph G_\times [10]:

$G_{\times} = G_1 \times G_2 = (V_{\times}, E_{\times})$, where $V_{\times} = \{(v_1, v_2) : v_1 \in V_1, v_2 \in V_2\}$, $E_{\times} = \{(v_1, v_2), (u_1, u_2)\} : \{v_1, u_1\} \in E_1 \wedge \{v_2, u_2\} \in E_2\}$. Following the theorem [47] that a single random walk on the direct product graph G_{\times} is equivalent to random walk on the two graphs G_1, G_2 at the same time, the P -step random walk kernel is computed by:

$$K(G_1, G_2) = \sum_{p=0}^P K_p(G_1, G_2) = \sum_{p=0}^P \lambda_p \sum_{i,j=1}^{|V_{\times}|} [A_{\times}^p]_{ij},$$

where A_{\times} is the adjacency matrix of the direct product graph, λ_p is the sequence of weights and the (i, j) item of A_{\times}^p represents the P -length common walks between the i -th and j -th nodes.

Extending the above definition to the scenario of graphs with node attributes, we use $\mathbf{X} \in \mathbb{R}^{n \times d}$ to denote the attribute matrix of a graph with n nodes and attribute dimension d . The node attribute matrices of G_1, G_2 are denoted by $\mathbf{X}_1 \in \mathbb{R}^{n_1 \times d}, \mathbf{X}_2 \in \mathbb{R}^{n_2 \times d}$. Thus, for the graph $G_{\times} = G_1 \times G_2$, the node attribute matrix is $\mathbf{S} = \mathbf{X}_1 \mathbf{X}_2^T, \mathbf{S} \in \mathbb{R}^{n_1 \times n_2}$, where the item (i, j) encodes the similarity between the i -th node in G_1 and the j -th node in G_2 . We flattened \mathbf{S} to $\mathbf{s} \in \mathbb{R}^{n_1 n_2}$ to obtain the formula for the random walk kernel:

$$K_p(G_1, G_2) = \sum_{i,j=1}^{|V_{\times}|} \mathbf{s}_i \mathbf{s}_j [A_{\times}^p]_{ij} = \mathbf{s}^T A_{\times}^p \mathbf{s}.$$

A.5 Experiment Details

A.5.1 Dataset Statistics and Sources. Statistics and sources of the datasets are shown in Table 6. The region division is based on census tracts and the urban imagery are 256 x 256 pixels with a resolution of 4.7 meters. For carbon emissions data provided in 1x1 km grid cells, we map them to our regional divisions. If a region covers more than one grid cell, the emissions from all relevant cells are summed. For the income and education data, each region is assigned a specific value representing the average income or number of people with an education for that region.

Table 6: Dataset Statistics and Sources.

Data	NYC	CHI	Source
Region	180	869	Census Bureau [4]
POI	143,445	112,989	OSM [38], Safegraph [40]
Imagery	7,160	-	ArcGIS [1, 31]
Taxi trip	12,684,353	3,889,032	NYCOD [36], CHIDP [6]
Road	110,919	139,850	OSM [38]
Junction	62,437	66,065	OSM [38]
Crime	35,335	274,733	NYCOD [36], CHIDP [6]
Check-in	106,902	-	Foursquare [12]
Crash	-	150,535	CHIDP [6]
Carbon	1x1 km	1x1 km	ODIAC [37]
Income	180	-	NYCOD [36]
Education	-	869	CHIDP [6]

A.5.2 Baselines Descriptions. Here we provide more details about the three types of baseline methods.

I. Knowledge graph embedding methods:

- **TransR** [27] projects entities and relationships into relation-specific spaces. **TransR-N** uses the region node embedding as the representation. **TransR-G** averages embeddings of all nodes in the region subgraph.

II. Graph embedding methods:

- **node2vec** [14] uses random walks to generate node sequences and learns representations with the Skip-Gram model.
- **GAE** [22] has an encoder for node representations and a decoder for reconstructing the adjacency matrix.

III. State-of-the-art urban region embedding methods:

- **MVURE** [64] constructs multi-view data based on human mobility and region attributes for region representation.
- **MGFN** [50] learns relationships among human mobility patterns through cross-attention.
- **HREP** [69] creates a multi-relational graph and a relation-aware GCN with a non-graph prefix prompt. **HRE** is a variant without the prompt module.
- **ReCP** [25] learns region representation through a multi-view fusion that captures unique and consistent information.

A.5.3 Implementation. We use five-fold cross-validation to evaluate GURPP and baselines on multiple tasks across both cities. In GURPP, the region embedding dimension is set to 144 [69]. The HGT layer and head numbers are 2 and 4, respectively; triplet margin $\delta = 2$; fusion loss weight $\mu = 0.01$; learning rate is 0.001; and weight decay is 1e-6. In the task-learnable prompt module, we set the random walk kernel layers to 2, prompt graph sizes to 6 and 8, and region subgraph node size to 50. Due to the lack of satellite image datasets for CHI, we removed the image encoding module and adjusted the fusion layer dimensions accordingly.

Our model is implemented with PyTorch 1.8 and DGL 2.2.1+cu118 on a machine with an NVIDIA RTX 3090 GPU, 64 GB RAM, and a 3.50 GHz CPU. Scalability experiments use an NVIDIA H100 80GB GPU, 1 TB RAM, and a 3.2 GHz CPU.

A.5.4 Comparison with Language Prompt. We conduct a comparison with LLMs, including Deepseek R1, Gemini 2 Flash, in predicting the top 10 regions with the highest crime probabilities in NYC. The specific variant of GURPP used in this comparison is GURPP_{M-P4}. We provide each language model with detailed descriptions of the 180 regions in NYC, including region IDs and information about POIs. The prompt is as follows.

I will provide information on the division of 180 regions in Manhattan, New York, with each region including a region ID and detailed description. Please analyze the potential crime risk of each area based on their descriptions, rank them according to the likelihood of criminal incidents from highest to lowest, and list the top 10 highest-risk regions. The regional information is formatted as: {*mhtr0*: *mhtr0 is located in Manhattan, encompassing 1318 POIs, with the main types being Restaurants and Other Eating Places, Clothing Stores ...*} ...